

UNIVERSIDAD AUTÓNOMA DE MADRID
ESCUELA POLITÉCNICA SUPERIOR



Grado en Ingeniería de Tecnologías y Servicios de Telecomunicación

TRABAJO FIN DE GRADO

SEGMENTACIÓN OBJETO-FONDO.

Paula Moral de Eusebio.
Tutor: Álvaro García Martín.
Ponente: José María Martínez Sánchez.

Julio 2017

SEGMENTACIÓN OBJETO-FONDO.

Paula Moral de Eusebio

Tutor: Álvaro García Martín

Ponente: José María Martínez Sánchez



Video Processing and Understanding Lab

Departamento de Tecnología Electrónica y de las Comunicaciones

Escuela Politécnica Superior

Universidad Autónoma de Madrid

Julio 2017

Trabajo parcialmente financiado por el Ministerio de Economía y Competitividad del Gobierno de España bajo el proyecto TEC2014-53176-R (HAVideo) (2015-2017)



Resumen

El objetivo principal de este trabajo de fin de grado es la realización de un segmentador objeto-fondo capaz de generar máscaras de segmentación en las que las imágenes se dividan en el elemento a segregar y el resto de ellas. Esto se ha llevado a cabo mediante la generalización del trabajo previo del que partimos, el segmentador persona-fondo, cuya característica principal es el empleo de detección para generar dichas máscaras. La diferencia a destacar entre ambos segmentadores es el modelo que utilizan para la detección. En el caso de persona es siempre el mismo con una sola pose, mientras que el segmentador objeto-fondo puede utilizar cualquier modelo con un número de poses indeterminado. Para el desarrollo de nuestro sistema se verán dos procedimientos de aplicación, primero mediante un método donde se buscan las partes independientes del modelo de objeto y luego otro método donde estas partes se combinen y la detección se haga mediante las partes dependientes. Finalmente, se comprobarán los resultados de ambos métodos y se realizará la evaluación del sistema y una comparación respecto al sistema original.

Para la obtención de las pruebas y la validación de nuestro algoritmo es necesario realizar una selección de secuencias de imágenes o *datasets* y de un conjunto de modelos de objetos a detectar que resulten útiles para su aplicabilidad a nuestro sistema.

Palabras clave

Segmentación objeto-fondo, segmentación persona-fondo, detección de personas, detección de objetos, partes independientes, partes dependientes, máscaras de segmentación.

Abstract

This work aims to develop an object-background segmenter capable of generating segmentation masks where the images can be divided into the element to be segregated and the remaining ones. This has been achieved through the generalization of the previous work from which we proceed, the person-background segmenter, whose main feature is the use of detection in order to generate such masks. The difference to be highlighted between the two segmenters is the model they use for detection. In the person case, the model is always the same with a single pose, while the object-background segmenter can use any model with an indeterminate number of poses. To accomplish our system, two application procedures will be analyzed in this work: one method in which the independent parts of the object model can be detected, and another one in which these parts could be combined. Finally, the results of both methods will be put to the test, and the system will be assessed.

To obtain the tests and validation of our algorithm it is necessary to conduct a selection of sequences of images or datasets and of a set of object models to detect whether they are useful for their application to our system or not.

Keywords

Object-Background segmentation, people-background segmentation, people detection, object detection, independent parts, dependent parts, segmentation mask.

Agradecimientos

Quería aprovechar este apartado para dar las gracias a mi tutor, Álvaro García, por darme la oportunidad de realizar este trabajo y ayudarme en todo momento que lo he necesitado. Quería también agradecer a mi familia el apoyo y cariño incondicional que siempre me han dado y la educación que me han ofrecido, gracias a la cual hoy puedo estar escribiendo estas líneas. Por último, agradecer a todos los amigos y compañeros que han estado presentes a lo largo de este camino y sin los cuales no habría llegado al final de la carrera.

Índice general

Resumen	v
Abstract	vii
Agradecimientos	ix
1. Introducción	1
1.1. Motivación	1
1.2. Objetivos	1
1.3. Organización de la memoria	2
2. Estado del arte	3
2.1. Introducción	3
2.2. Detección Objetos	3
2.2.1. Detector basado en modelos holísticos	4
2.2.2. Detector basado en modelos por partes	5
2.3. Segmentación persona-fondo	5
3. Diseño	9
3.1. Introducción	9
3.2. Punto de partida	9
3.3. Dataset	10
4. Desarrollo	13
4.1. Introducción	13
4.2. Algoritmo	13
4.2.1. Independent Object Part	14
4.2.2. Dependent Object Part	15
4.3. Selección del dataset	17
5. Integración, pruebas y resultados	21
5.1. Introducción	21
5.2. Métrica de evaluación	21
5.3. Pruebas y resultados	23

6. Conclusiones y trabajo futuro	29
6.1. Conclusiones	29
6.2. Trabajo futuro	30
Bibliografía	32
A. Glosario	35
B. Gráficas ROC y tabla AUC	37

Índice de figuras

2.1. Pirámide de características	6
2.2. Ejemplo de mapa de confianza	7
2.3. Representación de partes del cuerpo	7
3.1. Ejemplo de imagen y ground truth	10
3.2. Ejemplo de varios ground truth dentro de una secuencia	11
3.3. Ejemplo modelo	11
4.1. <i>Anchor</i> bicicleta	14
4.2. Particiones dependientes	16
4.3. Combinación de partes	17
4.4. Detección autobús	18
4.5. Detección caballo	19
5.1. Ejemplo ROC con distintos fp	22
5.2. Gráficas AUC	24
5.3. ROC para secuencia <i>Bmx Segtrackv2</i> distintos <i>sbins</i>	25
5.4. Máscaras de segmentación para imagen de <i>Bmx Segtrackv2</i>	26
5.5. Máscaras para IOP coincidentes con Original	27
5.6. Segmentación de secuencia vaca	28
5.7. Segmentación Persona-Fondo	28
B.1. Gráficas AUC	39

Índice de tablas

5.1. AUC Car turn	23
5.2. AUC de las secuencias ideales	25
B.1. AUC de todas las secuencias para $fp=1,2,4, 10$	38

Capítulo 1

Introducción

1.1. Motivación

La segmentación es una técnica empleada para la división de una imagen en regiones homogéneas con alguna característica en común, con el fin de simplificarla e identificar elementos de interés para una aplicación concreta. En este trabajo se plantea un enfoque distinto al tradicional, con el que se pretende identificar las zonas de la imagen con mayor probabilidad de no ser objeto (segmentación objeto-fondo), es decir, de pertenecer al fondo, mientras que en la segmentación convencional frente-fondo lo que se busca es la correcta división de la imagen en objetos del frente (*foreground*) del resto de objetos o fondo (*background*).

Este nuevo enfoque permite el desarrollo de aplicaciones de procesamiento automático de vídeo. Se puede usar tanto en pre-procesado como en post-procesado en algoritmos de *tracking*, en seguimiento y estimación de densidad de objetos, aplicaciones de robótica, de asistencia a la conducción, de video-seguridad, etc. Una de sus principales ventajas es su posibilidad de aplicación en distintos escenarios complejos y la independencia a movimientos de la cámara al analizar las secuencias *frame* a *frame*.

1.2. Objetivos

El principal objetivo del trabajo se centra en la segmentación de secuencias de imágenes para identificar las zonas de la imagen con mayor probabilidad de no ser objeto, es decir, de pertenecer al fondo. Por lo tanto, el tipo de segmentación que vamos a tratar en este trabajo es una segmentación orientada a determinar qué zonas de la imagen son fondo y qué zonas objeto, de modo que se le asignará una mayor

penalización a los píxeles clasificados incorrectamente como fondo cuando en realidad pertenecen al objeto. Es decir, la máscara final será con sesgo en el fondo y no en el objeto, como suele ser el caso de los algoritmos de segmentación y detección tradicionales.

Se partirá de un algoritmo de segmentación persona-fondo definido en [1] con las mismas características mencionadas, para poder generalizarlo con la finalidad de que no sea aplicable solo a personas, sino también a distintos objetos.

Los objetivos que se van a abordar son los siguientes:

1. El estudio del estado del arte, en el que se realizará un análisis de toda la tecnología ya existente relacionada con este trabajo.
2. El estudio del algoritmo original segmentación persona-fondo con distinto error de coste.
3. Adaptación de dicho algoritmo original segmentación persona-fondo con el objetivo de generalizar para distintos elementos.
4. Estudio de los diferentes *datasets* y modelos de objetos disponibles para su aplicabilidad a nuestro sistema.
5. Evaluación de los resultados obtenidos.

1.3. Organización de la memoria

La memoria consta de los siguientes capítulos:

- Capítulo 1 Introducción: motivación y objetivos.
- Capítulo 2 Estado del arte: Detección de objetos y segmentación.
- Capítulo 3 Diseño: Punto de partida y *dataset*.
- Capítulo 4 Desarrollo: Algoritmo y selección del *dataset* de secuencias a usar.
- Capítulo 5 Integración, pruebas y resultados.
- Capítulo 6 Conclusiones y trabajo futuro.
- Bibliografía.

Capítulo 2

Estado del arte

2.1. Introducción

En este capítulo se va a realizar una visión general de la tecnología ya existente que guarda conexión con nuestro trabajo. El algoritmo en el que nos vamos a basar es el descrito en [1], en el que la segmentación persona-fondo se realiza mediante detección. En él se crea la máscara de segmentación a partir de los mapas de confianza de detección de partes del modelo, los cuales se combinan o no para generar la máscara de fondo que finalmente es procesada por operaciones morfológicas.

Por lo tanto, el desarrollo del Estado del Arte comenzará en la sección 2.2, en ella se verán los distintos detectores presentes en la tecnología empleada, los holísticos y basados en partes. En el apartado 2.2.2.1 se explicará el detector original DPM (*Deformable Parts Model*) [2] en el que se basa la segmentación persona-fondo con distinto error de coste [1], estando ésta explicada en la sección 2.3.

2.2. Detección Objetos

La detección es una de las líneas de investigación fundamentales en el procesamiento de imagen y vídeo, cuya finalidad es la de localizar los elementos en la escena. Presenta la dificultad de tener que reconocer personas u objetos de una categoría determinada, pudiendo pertenecer a ésta elementos que difieran mucho en apariencia. Un ejemplo de dicho problema sería el de detección dentro de la categoría de coche de varios vehículos de distintos modelos, colores, tamaños, etc. Por lo tanto, no solamente interviene en la problemática de detección la variación de iluminación y el punto de vista, sino que también lo hacen las diversas variaciones en las categorías

politéticas¹, naciendo así detectores de objetos capaces de localizar objetos variables utilizando una mezcla de partes en las que se dividen los modelos [2].

Distinguiremos dentro de los detectores dos grupos, los holísticos y los basados en partes que veremos con más detalle en las secciones 2.2.1 y 2.2.2. Los modelos holísticos tratan el objeto en cuestión como un todo, sin diferenciar partes dentro de la integridad de éste, mientras que los basados en partes dividen el modelo en varias secciones. Así como los detectores basados en modelos holísticos tienen una menor complejidad que los basados en partes, éstos no funcionan correctamente con oclusiones parciales o variaciones de la posición mientras que los basados en partes sí.

2.2.1. Detector basado en modelos holísticos

Para todos los modelos de objetos que presentan deformaciones, como pueden ser los casos de personas o animales (éstos pueden tener variaciones de poses al no ser objetos rígidos), encontrar el modelo íntegro (*root*) no suele dar resultados satisfactorios. De la misma manera, las oclusiones de los elementos a encontrar en escenas del mundo real presentan también dificultades. Para lidiar con estos problemas se proponen algoritmos que obtengan las características de información de borde. Entre éstos destaca el algoritmo ISM (*Implicit Shape Model*) [3] que presenta un codebook y especifica en qué parte del objeto pueden ocurrir las entradas de éste. Destacan también los detectores holísticos que usan un enfoque basado en ventana deslizante definido con características Haar y un clasificador en cascada Adaboost [4], o definidos por Histogramas de Gradientes Orientados HOG [5].

- **ISM (*Implicit Shape Model*):** El algoritmo consta de un codebook con las apariencias locales prototípicas para las clases de objetos y la distribución de probabilidad espacial que especifica dónde puede encontrarse cada entrada del código en el objeto. No se intenta definir un modelo explícito de todas las posibles formas que pueden adoptar los elementos de la misma clase, sino que se definen formas con apariencias consistentes entre sí, lo que permite combinar información para reconocer alguna posición en concreto en lugar de tener que guardar todas las posibles articulaciones del modelo [3].
- **Características Haar y Clasificador Adaboost:** Algunos detectores holísticos se basan en un sistema de ventana deslizante definido con características Haar y un clasificador en cascada Adaboost [4]. Estas características se calculan en muchas escalas. Al haber más características Haar que número de píxeles,

¹Una categoría política aúna un gran número de características y no exige que todos los miembros de ésta posean todas las características, (aunque sí que tengan cierta homogeneidad en ellas).

para poder asegurar una clasificación rápida se tienen que excluir las características mayoritarias, quedando solamente aquellas que sean más críticas. La selección de dichas características se realiza mediante el algoritmo de aprendizaje Adaboost.

- **HOG (*Histograms of Gradients*):** El método de Histograma de Gradientes Orientados realiza la evaluación de histogramas locales normalizados de las orientaciones de los gradientes de la imagen. La apariencia y forma de objetos locales se pueden caracterizar por gradientes locales de intensidad o por la dirección de sus bordes [5]. La forma práctica de realizar esto es mediante la división de las ventanas de la imagen en regiones o cells, en las que se calculará para los píxeles contenidos en éstos el histograma de las direcciones de gradiente o de las orientaciones de borde. Se realiza una acumulación de la energía de los histogramas locales sobre regiones mayores que los cells denominadas *blocks*, lo que normaliza el contraste con la finalidad de lograr una mejora ante la luminosidad, sombras, etc. Estos *blocks* normalizados serán los descriptores HOG.

2.2.2. Detector basado en modelos por partes

Como se ha mencionado en la Introducción, los modelos basados en partes dividen el elemento en regiones de manera que, aunque hacen el algoritmo más complejo, evitan el problema de oclusiones y variaciones de posturas.

2.2.2.1. Discriminatively Trained Deformable Part-based

DPM (Deformable Parts Model) es un detector de objetos basado en la mezcla de modelos por partes deformables multiescala, definido en [2]. El detector viene especificado por filtros que se aplican a la pirámide de características HOG, la cual contiene el mapa de características de sucesivas escalas. El sistema aplica el filtro *root* (define el objeto íntegro) o los filtros para cada parte del modelo de mayor resolución.

El proceso que sigue el DPM comienza por sacar el mapa de características y el mapa de características al doble de resolución de cada escala. En el primero se aplica el filtro *root* y en el segundo los filtros por partes. Las repuestas transformadas se combinan y dan como resultado una puntuación final para cada localización del *root*.

2.3. Segmentación persona-fondo

La segmentación en la que nos vamos a basar en este trabajo es la definida en [1]. Ésta realiza la separación de la imagen en dos clases diferentes, persona y fondo, bus-

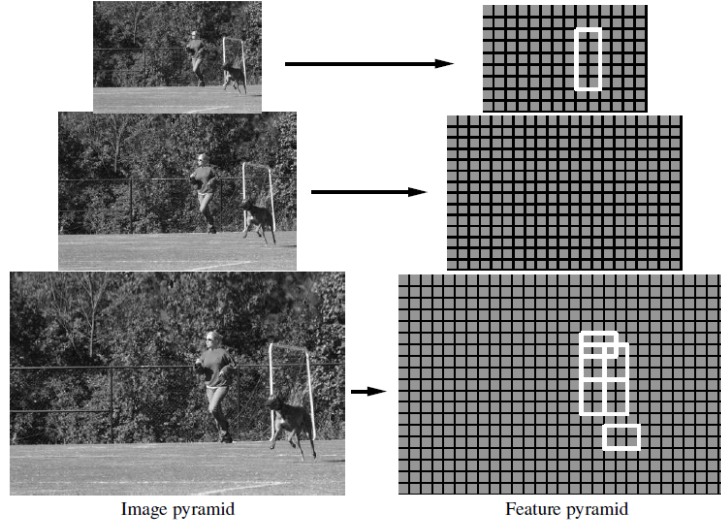


Figura 2.1: Pirámide de características en diferentes escalas. Los filtros por partes se sitúan en una resolución espacial el doble a la que se encuentra el *root*. Figura extraída de [2]

cando que ninguna persona o parte del cuerpo se clasifique erróneamente como fondo. Si sucede que el fondo se detecta en la clase incorrecta, tiene menor penalización, por lo que se dice que es una segmentación con sesgo en el fondo.

Se va a realizar una combinación de la respuesta de detectores holísticos y basados en partes para formar un modelo de verosimilitud conjunta que incluya casos de personas con oclusiones. En concreto este algoritmo se basa en el detector DPM [2] explicado en 2.2.2.1. Los mapas de confianza (o máscaras de segmentación) resultantes de la detección de personas servirán para localizar las áreas de la imagen donde no se encuentre ninguna persona.

Dentro de este sistema hay cinco métodos para la representación de partes del cuerpo (figura 2.3) : IBP (Independen Body Parts), DBP (Dependet Body Parts), sus versiones extendidas IEBP y DEBP y la versión post-procesada DEBP-P. No se va a incidir en estas últimas versiones IEBP, DEBP y DEBP-P ya que no están contempladas en el objetivo principal de este trabajo, que es la generalización del segmentador para cualquier objeto, quedando así como perspectiva de trabajo futuro.

Para el detector multiescala basado en partes, $P_n(x, y, s)$ representa la confianza en la posición de píxel (x, y) para la parte del cuerpo n asociada a la escala s . Cada parte del cuerpo está modelada por una 3-tupla $(F_n, v_{n,0}, d_n)$ donde F_n es la respuesta al filtro HOG [5] para la parte n , $v_{n,0}$ es un vector de dos dimensiones que define la

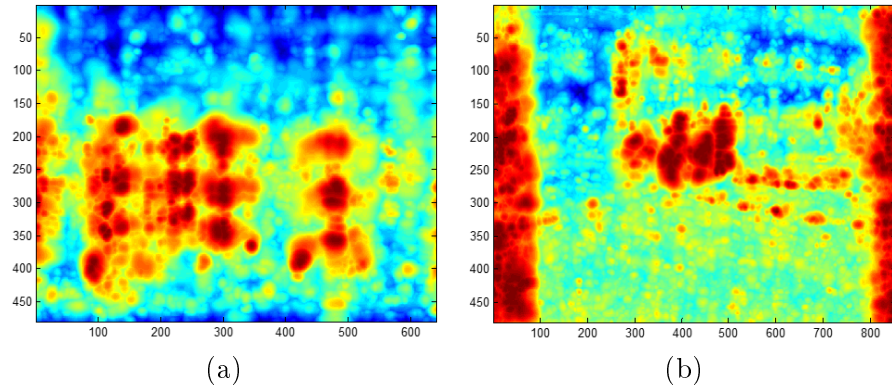


Figura 2.2: Mapa de confianza de Persona (a) y mapa de confianza de coche (b)

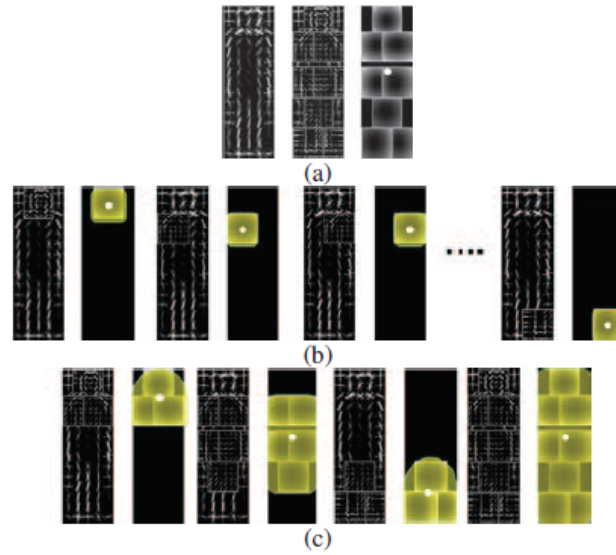


Figura 2.3: (a) Modelo de múltiples partes de persona de [2]; (b)Modelo IBP; (c) Modelo DBP.

posición relativa de la parte respecto a la posición del *anchor* (x_0, y_0) y d_n es un vector de cuatro dimensiones que define el coste para cada posible localización de la parte del cuerpo relativo a la posición del *anchor*, lo que otorga al algoritmo una mayor distorsión de las partes.

La puntuación para la confianza de la parte n en la escala s viene dada por:

$$P_n(x, y, s) = F_n(x, y, s) \cdot \langle d_n, \Phi(dx_n, dy_n) \rangle$$

donde (dx_n, dy_n) indican el desplazamiento de la parte n relativa *anchor*.

Se emplean ocho partes independientes ($N = 8$) del cuerpo para definir el método IBP, y se define la posición relativa del *anchor* respecto a la parte n correspondiente en lugar de definirla relativa al *root*. Para definir el método DBP, el cual es más robusto que IBP, se usan cuatro modelos ($M = 4$) de partes dependientes las cuales son combinaciones de las partes independientes. Se escoge este número de modelos debido a la correlación entre las partes de la persona: cuerpo entero (*root*), cabeza-hombros, tronco y pies. Se propone extender la definición de partes dependientes para utilizar la información de otras partes del cuerpo dependientes con la finalidad de recuperar partes del cuerpo no detectadas o mejorar la confianza entre las partes del cuerpo ya detectadas. Cada parte del cuerpo extendida viene dada por el máximo entre la parte dependiente original y el promedio de las otras partes del cuerpo dependientes relativas a la misma parte dependiente original.

Si se asume que debe haber al menos dos partes dependientes del cuerpo visibles, se puede recuperar el resto de la persona haciendo un promediado de las partes restantes, evitando así la recuperación de detecciones incorrectas aisladas.

La evaluación del rendimiento del segmentador se mide en términos de AUC (*Area Under the Curve*) con penalización de diferentes falsos positivos, en concreto $f_p = 1, 2, 4, 10$. El valor $f_p = 1$ equivale a un segmentador tradicional frente-fondo, mientras que valores más altos de f_p dan mayor penalización a los resultados en los que se han detectado erróneamente partes de la persona como fondo, por ejemplo, un $f_p = 2$ corresponde al doble de penalización que para el caso $f_p = 1$.

Capítulo 3

Diseño

3.1. Introducción

En este capítulo de Diseño realizaremos un análisis previo al desarrollo del segmentador objeto-fondo, con la finalidad de poder determinar los pasos que se seguiremos y conocer las características del *dataset* necesarias para la explicación del desarrollo del trabajo realizado.

Definiremos el punto de partida, es decir, las funciones del algoritmo original que van a necesitar modificación para nuestro objetivo.

3.2. Punto de partida

Como hemos visto en la sección 2.3 del Estado del Arte, el segmentador persona-fondo [1] del que parte nuestro trabajo utiliza detectores holísticos y basados en partes para realizar la división de la imagen en ambas clases. En él se genera la máscara de segmentación persona-fondo a partir de la combinación de los mapas de confianza de detección de partes del modelo. Lo que se pretende es generalizar dicho algoritmo de modo que no aplique solo a personas, sino que lo haga también a cualquier tipo de objeto.

El esquema a seguir va a ser el mismo que en [1]. Primero se van a cargar las secuencias de imágenes en Matlab, de las cuales se empleará también su *ground truth* y a continuación, se cargará el objeto del modelo a buscar. Las funciones que se usarán son las mismas que en el algoritmo de segmentación persona-fondo con distinto error de coste salvo las que realizan el método IBP (*Independent Body Part*) y DBP (*Dependent Body Part*). Para el método de partes independientes IBP se recorrerá para cada imagen las tres poses y sus *flips* incluidos en el modelo del objeto y se generará

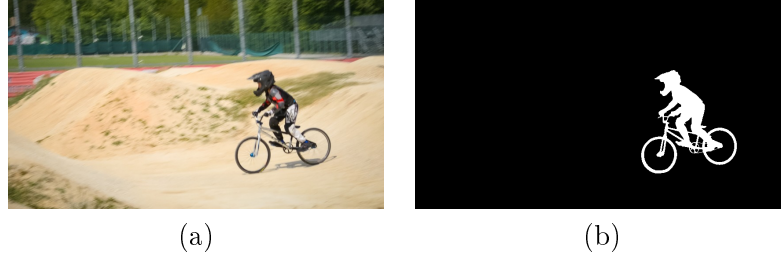


Figura 3.1: Ejemplo de imagen (a) y ground truth (b) en DAVIS.

la máscara de segmentación. En el caso del método de partes dependientes DBP, se realizará una combinación de las partes independientes de cada pose del modelo para realizar la máscara de segmentación. El resto de funciones, al ser independientes al número de poses, no van a necesitar modificaciones.

3.3. Dataset

El *dataset* (conjunto de elementos que se utilizan a la hora de evaluar el algoritmo) que se ha empleado en este trabajo está formado por un conjunto de secuencias de vídeos DAVIS [6] y SegTrackv2 [7].

DAVIS (*Densely Annotated Video Segmentation*) es un *dataset* específicamente diseñado para cubrir los retos propios de la segmentación de objetos en vídeos. En él aparecen un total de cincuenta secuencias de distintas longitudes cada una, las cuales comprenden distintas categorías: vehículos, personas practicando distintos deportes y animales. Cada secuencia tiene su correspondiente *ground truth* (máscara de segmentación con precisión de píxel de la imagen), y tanto éste como las imágenes están disponibles para las resoluciones 480p y 1080p.

El *dataset* SegTrackv2 ideado para segmentación mediante *tracking* consta de 14 secuencias de categorías de animales, vehículos y personas. A diferencia de DAVIS, cada secuencia tiene una resolución distinta. Para las secuencias en las que aparecen varios objetos detectables (por ejemplo, persona y bicicleta) están guardados los dos *ground truth* correspondientes, mientras que en DAVIS sólo se guarda uno con la persona y la bicicleta juntas (figuras 3.1 y 3.2).

Como ya hemos comentado, el objetivo es la realización de segmentación mediante detección y los modelos que se van a usar para ello vienen definidos en el algoritmo original DPM [8]. Éstos son los extraídos mediante el entrenamiento con secuencias del *dataset* de “PASCAL VOC 2009 challenge” [9] y se caracterizan por ser mezclas de tres componentes de modelos (tres poses) y sus respectivos flips (misma pose pero invertida). Un ejemplo de estos modelos es el de la figura 3.3. En ella aparecen para

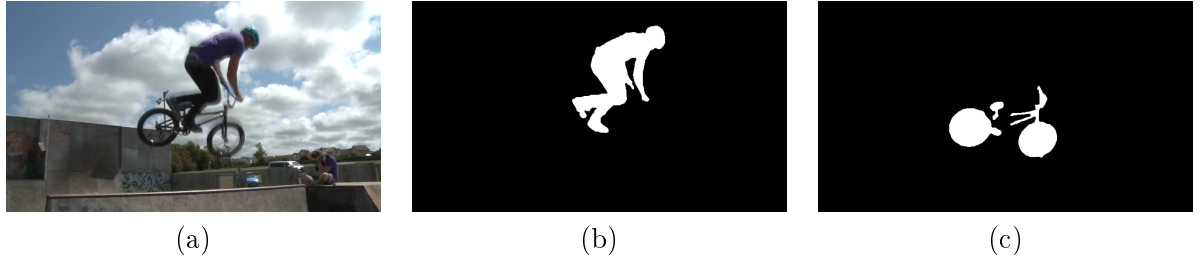


Figura 3.2: Ejemplo de imagen (a) y ground truth (b) y (c) en SegTrackv2.

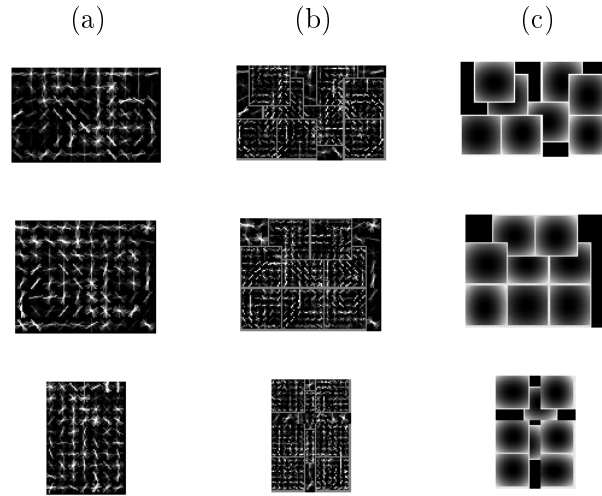


Figura 3.3: Modelo de bicicleta para tres posiciones distintas. Cada posición presenta: filtro *root* (a), filtros por partes de mayor resolución (b), modelo espacial de la ubicación de cada parte respecto al *root*(c).

cada posición del elemento bicicleta el filtro *root* (a), los filtros por partes (siempre son ocho particiones) de mayor resolución (b) y un modelo espacial de la ubicación de cada parte relativa al *root* (c). Cada componente del elemento tiene a su vez su *flip*, que es la misma pose pero invertida en el eje vertical (el objeto visto desde su parte izquierda o derecha al ser bilateralmente simétricos), por lo que se mirarán en total un conjunto de seis poses por modelo.

Capítulo 4

Desarrollo

4.1. Introducción

En este capítulo se van a explicar los cambios necesarios que se han llevado a cabo para la generalización del algoritmo de segmentación persona-fondo [1]. Además se describirá la selección de las secuencias del *dataset* que se podrán utilizar para la evaluación de esta tecnología.

4.2. Algoritmo

Una de las principales diferencias que se observan al analizar el algoritmo original persona-fondo en el que se basa nuestro trabajo es que el modelo del objeto a buscar no va a ser siempre el mismo. Al no ser éste un modelo fijo, el número de poses no tiene por qué serlo tampoco. Se toma la premisa de que éstas pueden variar, aunque en el caso de los elementos seleccionados en el capítulo anterior, todos están formados por tres poses y sus respectivos *flips*.

Como se explicó en el apartado 2.3 del Estado del Arte, los métodos en los que se calculan los mapas de confianza para la segmentación son IBP (*Independent Body Part*) y DBP (*Dependent Body Part*). IBP calcula la segmentación en función de las partes del cuerpo de manera independiente, mientras de DBP combina las partes para realizar este cálculo. En nuestro caso, generalizaremos ambos métodos llamándolos IOP (*Independent Object Part*) y DOP (*Dependent Object Part*).

Para estos métodos es necesario determinar la escala de la imagen a la que se va a buscar el objeto, que dependerá del tamaño de éste, siendo *sbins* mayores (*sbins* = 16) para objetos grandes y *sbins* menores (*sbins* = 4) para objetos pequeños. Se indicará también si el objeto cambia de tamaño mediante la variable *scale_levels*

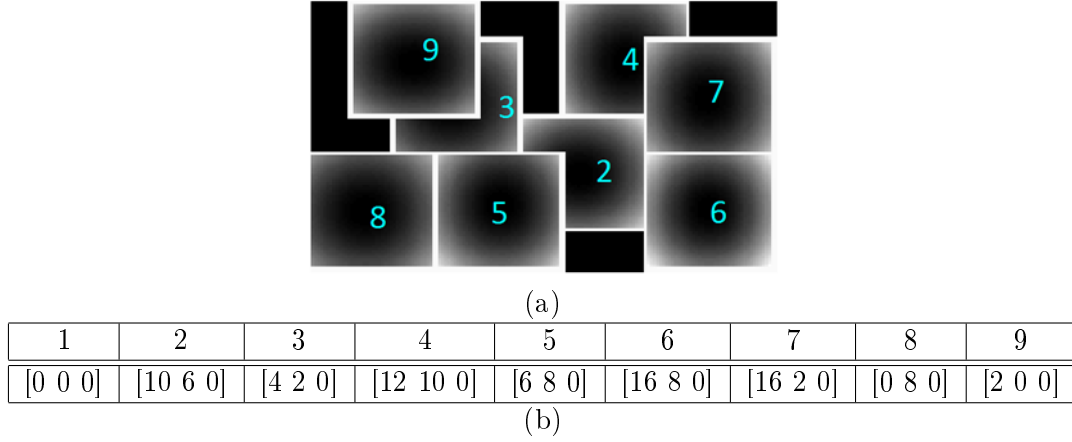


Figura 4.1: (a) Modelo de la bicicleta con el índice del *anchor* correspondiente (b) Tabla con los *anchors*

($scale_levels=[1\ 2]$) o se mantiene constante ($scale_levels=[1]$).

El resto de funciones originales se utilizan íntegras, ya que no necesitan ningún cambio al no involucrar en ellas el aumento de número de poses.

4.2.1. Independent Object Part

La idea principal del método IOP es la de detectar cada una de las partes que componen el objeto de manera independiente. A diferencia del algoritmo original DPM [2] que concentraba la puntuación o *score* obtenido por cada parte en un centro común del objeto, en nuestro caso buscamos justo lo contrario, que cada parte del cuerpo puntúe en su situación relativa a dicho centro, y no en el centro en sí.

Para ello, y para poder hacerlo de forma genérica a cualquier objeto, lo primero que debemos hacer es recorrer para cada imagen y su *ground truth* todas las poses y sus respectivos *flips*. De cada parte independiente (o bloque) dentro de la pose se va a calcular el *score*, el cual se realiza en una función ya implementada por el algoritmo original, pero es necesario hacer unas modificaciones en el modelo que va a recibir esta función. La posición del bloque relativa a cada pose del modelo viene determinada por su *anchor*, es decir, el *anchor* es la posición relativa de la parte independiente respecto al objeto. Es necesario concatenar el *anchor root* (de todo el objeto) al *anchor* de la sección en concreto de la que se esté calculando su *score*, para después restarle la distancia relativa necesaria al centro del objeto. Para realizar esta resta, calculamos la posición de la esquina superior izquierda del bloque (los bloques son cuadrados siempre definidos de ancho seis) y la posición resultante de esta esquina al centrar el bloque en el modelo. Una vez hecho esto se realiza el cálculo del *score*.

En el ejemplo de la figura 4.1 tenemos el modelo de partes independientes (cada bloque) del elemento bicicleta. Sobre cada parte hay un número que indica el índice de la tabla (b). Esta tabla contiene los *anchors* de cada bloque, los cuales indican la posición relativa de la parte respecto de la esquina superior izquierda del objeto. Se puede observar que el índice primero no aparece dibujado en la imagen, esto es debido a que son las coordenadas del elemento íntegro sin particiones. Si por ejemplo seleccionamos la parte del objeto número nueve, vemos que tiene *anchor* [2 0 0]. Para calcular el *score* relativo a este bloque se tiene que calcular el desplazamiento respecto del centro en el eje vertical (el eje horizontal no va a influir) por lo que, conociendo las dimensiones del modelo y sabiendo las dimensiones de cada bloque se calcula este desplazamiento y se modifica el *anchor*.

4.2.2. Dependent Object Part

Una forma más efectiva del cálculo de los mapas de confianza es mediante la combinación de partes. La finalidad de hacer ésto es asegurar que, si se detecta una parte se va a detectar el resto del objeto, no quedando así ninguna zona sin cubrir. Para ello, se van a agrupar las partes independientes en cuatro conjuntos (se han escogido cuatro al haber siempre ocho particiones en cada modelo, y ser el número de conjuntos usado en el algoritmo original [1]) y se van a calcular los *scores* dependientes de cada conjunto respecto del resto de combinación de partes.

Como la agrupación de las particiones del algoritmo original se hizo manualmente basándose en la relación que podían compartir las partes independientes (las partes superiores cabeza-hombros, las partes del tronco central hombros-brazos-entrepierna y por último las partes inferiores pies-entrepierna), la manera de seleccionar las nuevas secciones, en la que no se tiene por qué conocer la distribución de las particiones en el modelo, tiene que ser automática. Para ello se hace uso del algoritmo de *clustering Fuzzy C-means* (en concreto, su implementación para Matlab), que va a realizar las agrupaciones permitiendo que un mismo bloque esté incluido en varios grupos, tal como el algoritmo original.

4.2.2.1. Selección de agrupaciones

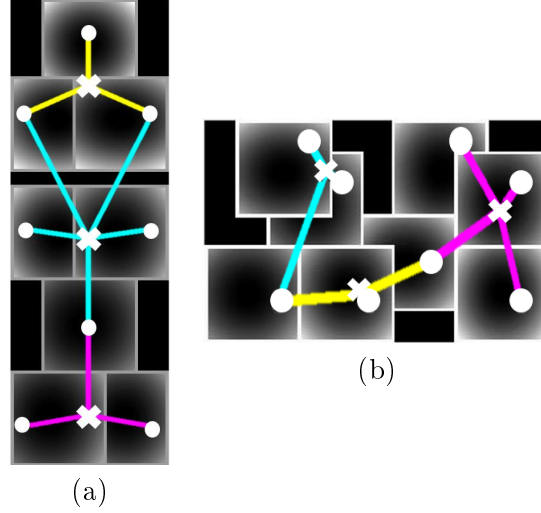


Figura 4.2: Particiones dependientes del modelo INRIA (a) y del modelo Bicicleta (b)

Se va a extraer la información relativa al *anchor* de todos los bloques para cada pose y su *flip* y, sabiendo el número de grupos o conjuntos en los que se quiere distribuir el modelo, se va a realizar el *clustering* de dichas coordenadas. Como ya se ha mencionado antes, serán cuatro conjuntos (uno de ellos será el elemento en su totalidad). *Fuzzy c-means* devuelve los centros de cada conjunto y la probabilidad de pertenencia de cada bloque a cada conjunto. Una vez halladas estas probabilidades, se podría seleccionar la máxima para asignar cada parte independiente a un grupo, pero como se busca que cada clúster tenga al menos un bloque en común con otro clúster y con este método no se garantiza, necesitamos otra condición que indique la pertenencia. Para ello, se va a hacer un barrido de probabilidades con un umbral que varíe de 1 a 0, de manera que éste vaya disminuyendo hasta que todos los conjuntos tengan bloques en común con al menos otro, o en el caso de que sea muy poco restrictivo, que un mismo bloque pertenezca a los tres clústeres. En la figura 4.2 vemos el resultado visual a través de colores de las agrupaciones de los bloques de la forma descrita, siendo el centroide de cada grupo el definido por un aspa.

4.2.2.2. Combinación de partes

Una vez seleccionados los grupos compuestos de partes independientes, se va a realizar la combinación de éstas de forma similar a la descrita en el apartado 4.2.1, con la salvedad de que ahora, en lugar de hacerlo de manera independiente con respecto al centro de todo el objeto, lo harán por cada grupo y respecto al centro de cada uno

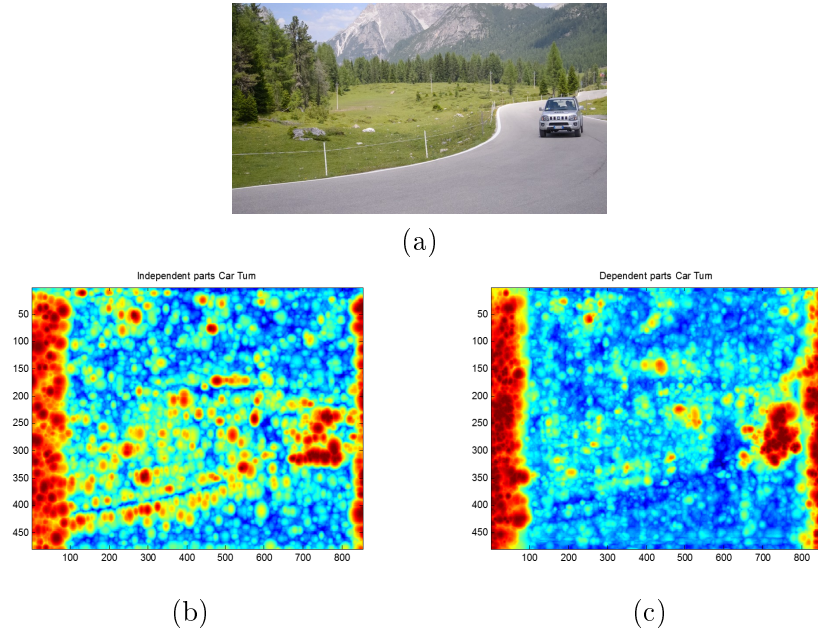


Figura 4.3: Combinación de partes: imagen original (a), máscara de segmentación IOB (b), máscara de segmentación DOB (c)

de los clústeres. Se va a realizar, para cada pose del modelo y su *flip*, un barrido para todas las agrupaciones de manera que se comparen con ella misma y con los otros tres grupos.

Finalmente se combinan todos los *scores* calculados, dando como ventaja una resultado mucho más preciso que en los casos anteriores.

4.3. Selección del dataset

Como se ha visto en el capítulo 3 de Diseño, los *datasets* escogidos contienen las secuencias de imágenes de distintas categorías de elementos y los modelos para detectar estos objetos. Para poder determinar qué imágenes nos van a servir para este trabajo, se han comparado las bases de datos de DAVIS [6] y SegTrackv2 [7] con los modelos de [8] y se han buscado los elementos que estuviesen incluidos tanto en éstos como en las secuencias de imágenes. Una vez hecha la comprobación de la coincidencia de todos los posibles modelos de objetos y secuencias de vídeo, se tiene un total de diez posibles objetos a detectar: bicicleta, pájaro, barco, autobús, coche, vaca, perro, caballo, motocicleta y tren.

Tras esta primera selección de *dataset*, en la que simplemente comparamos que los nombres de los archivos coincidieran con la categoría de objeto que se deseaba, se

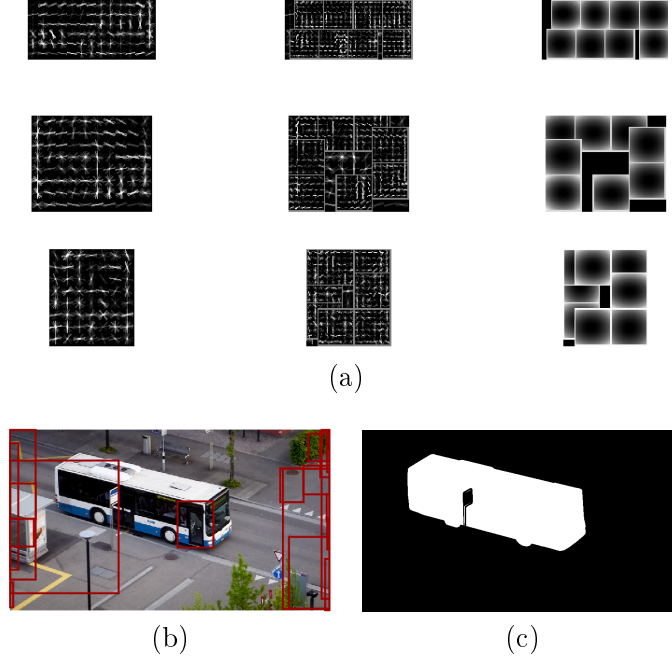


Figura 4.4: Detección de autobús. (a) Modelo de autobús, (b) Imagen con los bounding boxes de detección, (c) máscara del ground truth de segmentación.

va a realizar una segunda selección, esta vez más precisa, con el programa en Matlab donde está implementado el detector DPM [2]. Con este detector se quiere comprobar que se localizan los objetos en las secuencias donde aparecen. Para ello, únicamente introducimos las secuencias deseadas y modificamos el valor del *threshold* de la función para ampliar o disminuir el margen de restricción de detección (a mayor *threshold* menos restrictivo y más objetos con características parecidas son detectados como el elemento a buscar). De esta forma se hace evidente que para los modelos de pájaro, barco, autobús, perro y tren no se consigue detectar el elemento en las secuencias de imágenes para ningún valor del *threshold*. Ésto ocurre porque dichos elementos no cumplen con las características necesarias dentro de la categoría, es decir, la forma del objeto a buscar y la de la secuencia difieren mucho. Un ejemplo visual en el que falla la detección es el de la figura 4.4: se puede apreciar que con el modelo proporcionado no se detecta el autobús, ya que éste no guarda las similitudes necesarias (la imagen (b) es para un valor de *threshold* muy poco restrictivo, habiéndose realizado comprobaciones con distintos valores). En la figura 4.5 tenemos un caso en el que sí se ha detectado el elemento en cuestión, en este caso un caballo.

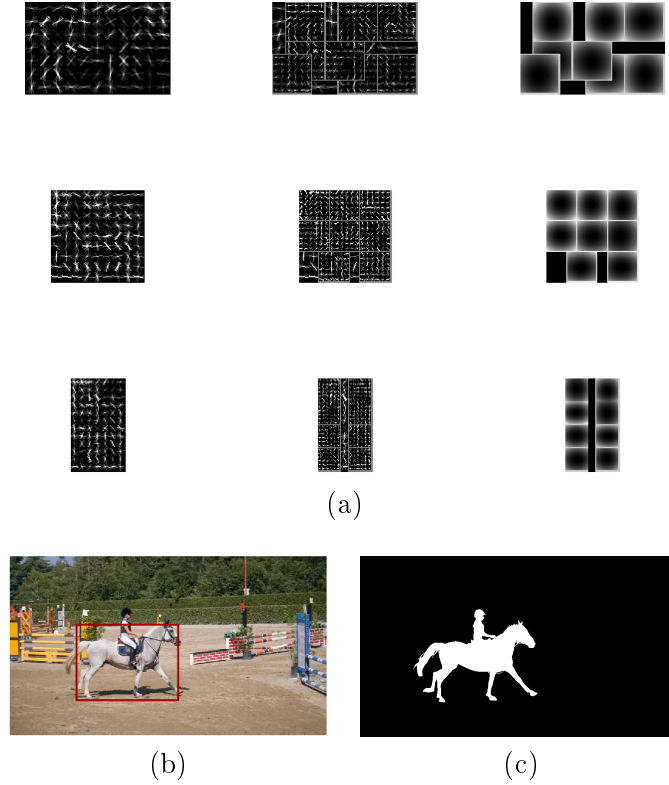


Figura 4.5: Detección de caballo. (a) Modelo de caballo, (b) Imagen con los bounding boxes de detección, (c) máscara del ground truth de segmentación

Finalmente, tras la selección visual de las secuencias coincidentes con los respectivos modelos y la selección más determinista de la aplicación del DPM en las imágenes, obtenemos un subconjunto del *dataset* original que nos va a ser útil para este trabajo. Dicho subconjunto se forma a partir de cinco modelos de objetos (coche, motocicleta, caballo, vaca, bicicleta), a detectar en un total de 12 secuencias. Además, se comparará en el capítulo 5 el algoritmo persona-fondo y objeto-fondo para el modelo de INRIA y las secuencias usadas en el algoritmo original, con lo que se pretende comprobar que el nuevo algoritmo no empeore el original al introducir los cambios necesarios.

Capítulo 5

Integración, pruebas y resultados

5.1. Introducción

El objetivo principal de este capítulo es el de evaluar el algoritmo de segmentación Objeto-Fondo con todas las secuencias necesarias para comprobar su funcionamiento y poder mostrar los resultados obtenidos. Para su realización, se va a definir la métrica de evaluación con la que observar y comparar la salida del algoritmo y se van a representar las máscaras de segmentación obtenidas por *frame* que sean más características para la explicación de los resultados. Las máscaras de segmentación se representan en una escala de colores de azul a rojo, de manera que para píxeles de la imagen donde esté el objeto éstos serán más rojos, donde haya falsas detecciones variará del rojo al amarillo, y donde no detecte nada de objeto los píxeles serán azules.

5.2. Métrica de evaluación

Para la evaluación de este algoritmo se va a emplear el área bajo la curva AUC (*Area Under the Curve*) definida por la ROC (*Receiver Operating Characteristic*) [3]. Ésta gráfica representa la capacidad de detectar los casos positivos correctamente (verdaderos positivos), frente a los resultados positivos incorrectos (falsos positivos). En el eje vertical se representará la sensibilidad, siendo ésta equivalente a los verdaderos positivos, y en el eje horizontal $1 - \textit{especificidad}$, siendo equivalente a los falsos positivos. El área contenida bajo la curva determina el rendimiento, estando directamente relacionados de manera que el aumento del área va a conllevar unos mejores resultados.

Los posibles resultados del AUC variarán entre cero y uno, cuanto más altos sean éstos, mejor será la evaluación que se esté realizando. En el caso concreto de nuestro

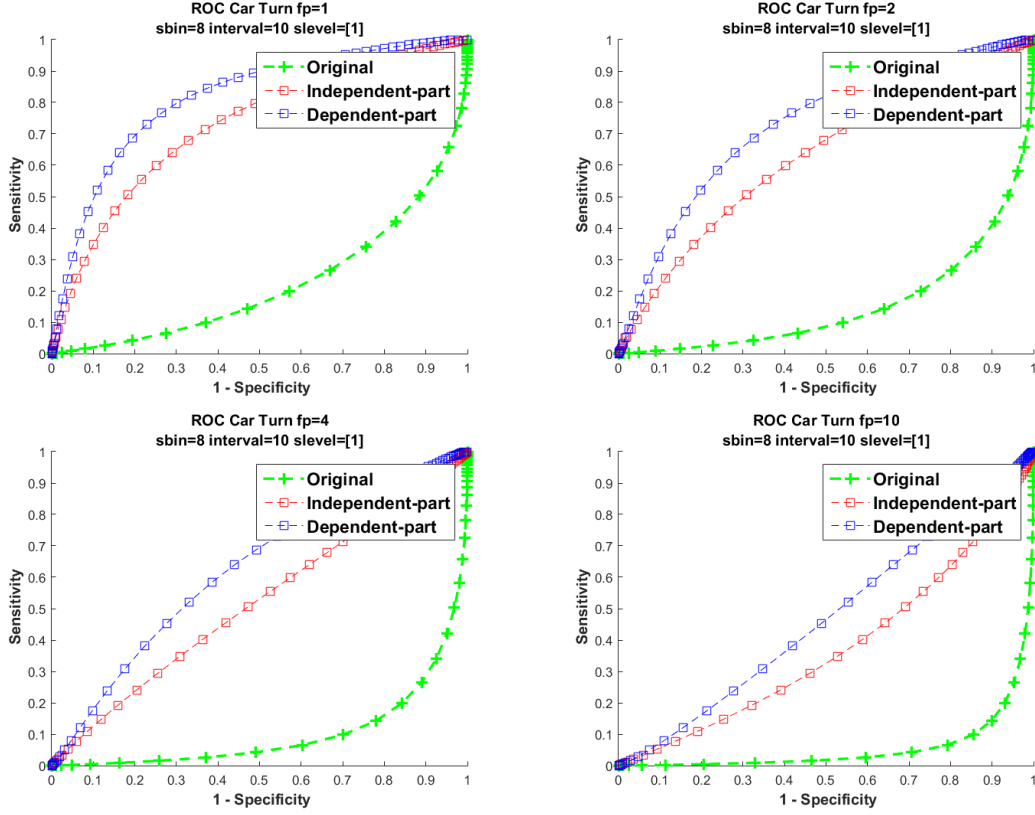


Figura 5.1: Ejemplo ROC de la secuencia “Car turn” de DAVIS para distintos factores de penalización de falsos positivos (fp).

trabajo queremos comprobar en qué medida la clasificación por píxel de cada secuencia se ha realizado correctamente.

Se van a comparar en una misma gráfica los resultados obtenidos de ejecutar cada secuencia en la función original de detección, en la función IOP de segmentación mediante detección de partes independientes y en la función DOP de segmentación mediante detección de partes dependientes. Esto se realizará mediante una función de Matlab facilitada por el VPU, implementada por el algoritmo original en la que sólo se tendrá que introducir los resultados de las secuencias de nuestro *dataset* y ésta calculará el AUC para distintos factores de penalización de los falsos positivos, en concreto $f_p = 1, 2, 4, 10$ (ver Apéndice B para los resultados del AUC). Como vemos en el ejemplo de la figura 5.1, el aumento del factor de penalización de falsos positivos conlleva una disminución de la ROC, por lo que se incrementa la penalización de los resultados en los que se han detectado erróneamente partes del objeto como fondo. Para esta secuencia en la tabla 5.1 vemos que, cuanto más bajo sea el número de factor de penalización de falsos positivos, el AUC tomará valores cercanos a uno.

Tabla 5.1: Valores AUC para la secuencia Car Turn

	Auc Original	Auc IOP	Auc DOP
Fp=1	0.2214	0.7275	0.8142
Fp=2	0.1535	0.6295	0.7303
Fp=4	0.1015	0.5207	0.6274
Fp=10	0.0551	0.3757	0.4740

Para DOP, los AUC son mayores que para IOP, y éstos a su vez mayores que para Original.

5.3. Pruebas y resultados

Como ya se ha mencionado, la representación del área bajo la curva va a ser el método para evaluar el correcto funcionamiento del algoritmo. De esta forma, el resultado deseable es un AUC que tome valores mayores a 0.5 y unas curvas ROC para las nuevas funciones IOP y DOP que no estén por debajo de la curva del detector Original. Esto es porque la función Original utiliza el algoritmo DPM [2] para realizar la detección con la que segmenta, de manera que sólo puntúa el centro del objeto, mientras que para IOP y DOP cualquier parte del objeto puntúa.

Se van a representar las curvas obtenidas tras la segmentación de cada una de las secuencias de nuestro *dataset* para distintos factores de penalización de los falsos positivos, en las que distinguimos tres grupos de comportamientos. Recogemos en la figura 5.2 tres casos característicos de cada comportamiento para el valor $fp = 1$ (ver Apéndice B para las gráficas de todas las secuencias). El primero lo formarían las secuencias *Car RoundAbout*, *Bmx Trees*, *Bmx Segtrackv2*, *Horse JumpLow* y *Car Turn*, cuyas ROC mantienen el comportamiento deseado descrito anteriormente (mayores para IOP y DOP que la curva de la función del detector original). El segundo comportamiento sería el de las secuencias *Bmx Bumps*, *Car Shadow*, *Drift*, *Motorbike*, y *Horse JumpHigh*, cuyos resultados no son tan ideales como los esperados. En estas secuencias los valores son muy cercanos a la ROC del original. Por último, para *Cow* y *Motocross Jump* los valores son muy cercanos a valores de 0.5 y del original. A continuación veremos con más detalle el comportamiento de cada grupo de secuencias.

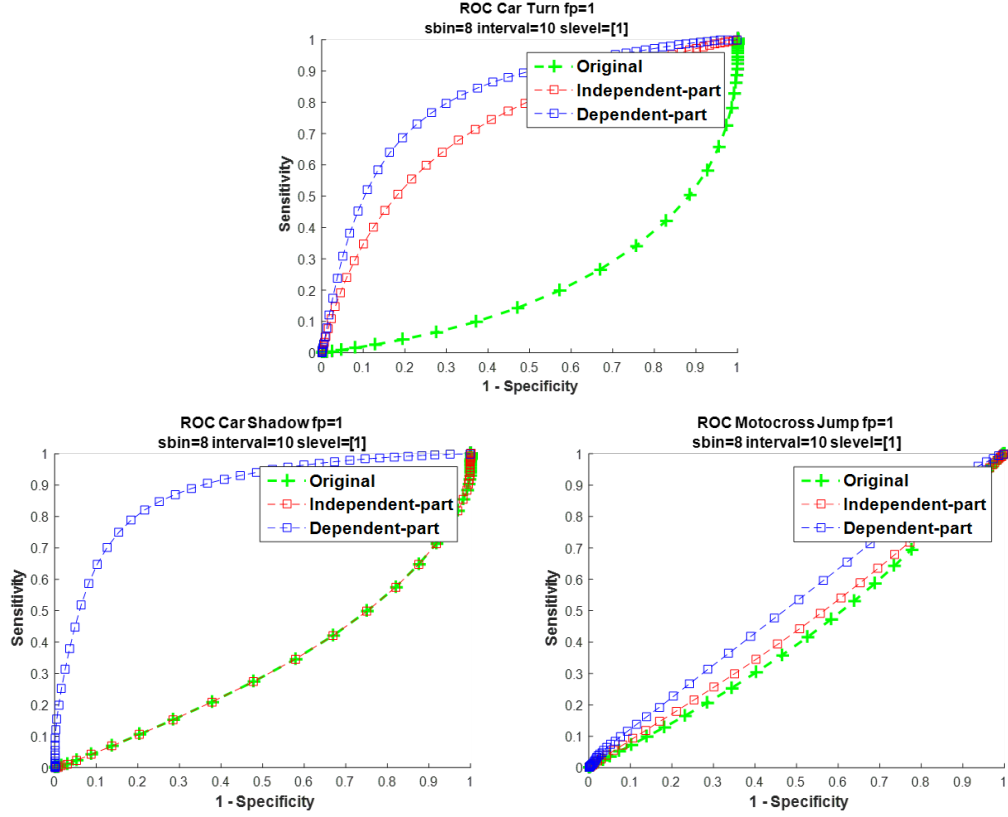


Figura 5.2: Gráficas AUC de las secuencias más características de cada grupo de comportamiento para un mismo valor fp .

- Tenemos recogidos los resultados ideales dentro de la tabla 5.2, donde se muestran los valores del área bajo la curva de cada secuencia. Podemos observar que, para IOP y DOP, todos los valores están por encima de 0.5 para factores de penalización de falsos positivos bajos, y la función Original da peores AUC que la función IOP, y ésta peores que la función DOP. Esto se debe a que, al hacer la función DOP uso de la combinación de partes independientes, se asegura la detección de todo el objeto con solo detectar una parte de él.

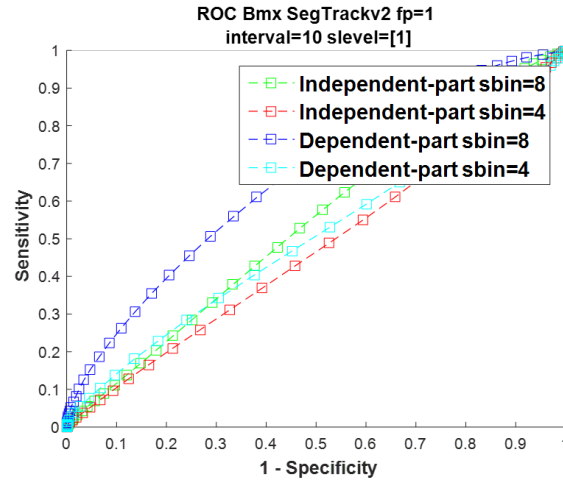


Figura 5.3: ROC para la secuencia *Bmx Segtrackv2* del segundo grupo de comportamiento con distintos valores de *sbins*

Tabla 5.2: Valores AUC para las secuencias que muestran los resultados ideales para $fp=1$ y $fp=10$ según la función Original, IOP y DOP.

	AUC Original		AUC IOP		AUC DOP	
	fp=1	fp=10	fp=1	fp=10	fp=1	fp=10
Car RoundAbout	0.3088	0.0866	0.6355	0.2770	0.7856	0.4926
Bmx Trees	0.5299	0.1971	0.5693	0.2038	0.7964	0.4479
Bmx Segtrackv2	0.4721	0.1787	0.5438	0.1966	0.6652	0.3080
Horse JumpLow	0.6088	0.2922	0.6892	0.3438	0.6868	0.3386
Car Turn	0.2214	0.0551	0.7275	0.3757	0.8142	0.4740

- Para el segundo grupo de comportamiento de resultados, aunque siguen siendo valores del AUC por encima del 0.5, los relativos a la función IOP coincide con los del original por lo que ya no son tan ideales. Para comprender mejor qué sucede en estos casos, cogemos una de estas secuencias para ejemplificar este comportamiento. Con la finalidad de descartar posibles problemas de detección con la escala en la que se busca el objeto, cambiamos ésta a una menor mediante un $sbin=4$ y, como vemos en la figura 5.3 de la secuencia *Bmx Segtrackv2*, los resultados disminuyendo el $sbin$ no solo no mejoran, sino que empeoran. Para terminar la comprobación, sacamos las máscaras de segmentación de una imagen de *Bmx Segtrackv2* (figura 5.4) para la segmentación por partes independientes con un $sbin=4$ y con un $sbin=8$. Al igual que vimos con la figura 5.3, no se mejora la detección disminuyendo la escala. Finalmente comparamos el resultado de la máscara de segmentación para una imagen de la secuencia de *Car Shadow* (en dicha secuencia coincidía la curva ROC relativa a la función IOP con la original)

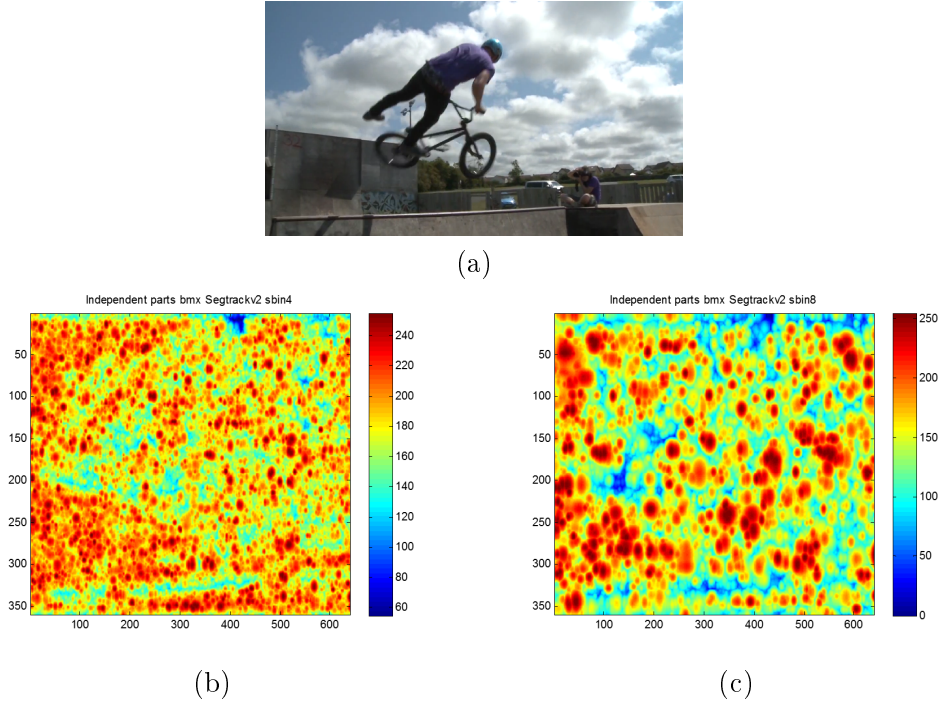


Figura 5.4: Imagen original de la secuencia *Bmx Segtrackv2* (a) y sus máscaras de segmentación para distintos *sbins* (b) y (c) .

con el resultado de la máscara de segmentación para una imagen de la secuencia *Car Turn* (con resultados óptimos). Como podemos observar en la figura 5.5, se obtiene para la detección con las partes independientes IOP una máscara de segmentación con más ruido en el caso de *Car Shadow* que para el de *Car Turn*.

- Respecto al último grupo, con la finalidad de comprender el comportamiento del segmentador en estas secuencias, se modificó el tamaño de la escala mediante el *sbin* y se observaron las máscaras de segmentación para comprobar si su detección mejoraba. Al no dar estas pruebas unos resultados favorables, llegamos a la conclusión de que, a pesar de haber probado el detector en una fase previa (explicado en la sección 4.3 del capítulo de Desarrollo) para determinar si nos servían las secuencias, en estos casos no van poder ser útiles para generar las máscaras de segmentación. En el caso particular de la secuencia *Cow*, se consideró como válida porque, aunque se detectaban erróneamente otras zonas de la imagen para todos los *threshold* (siendo éste más restrictivo o más laxo) se detectaba también la vaca, pero se ha comprobado que la máscara resultante no es la deseada (figura 5.6).

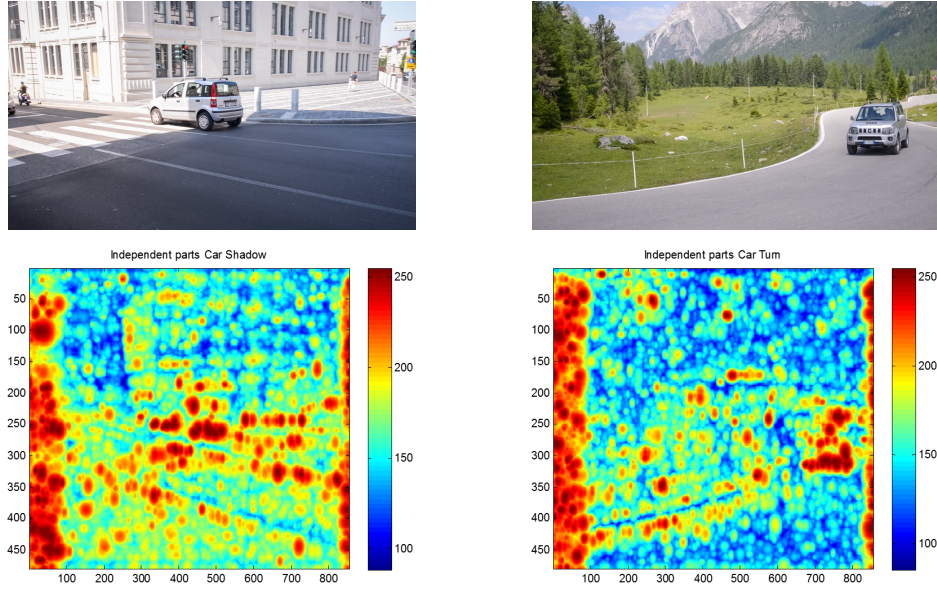


Figura 5.5: Máscaras para la función IOP coincidente con la función original en el caso de *Car Shadow* comparado con *Car Turn*.

Por último, para comprobar que el algoritmo genérico propuesto que hemos desarrollado no empeore los resultados del algoritmo original, hemos comparado los resultados que ambos generan empleando para ello el modelo INRIA y las secuencias de personas que se empleaban en la evaluación del sistema original. Se observa claramente en la figura 5.7 que las ROC resultantes de ambos algoritmos son prácticamente iguales, con valores de AUC cercanos a uno para todos los casos.

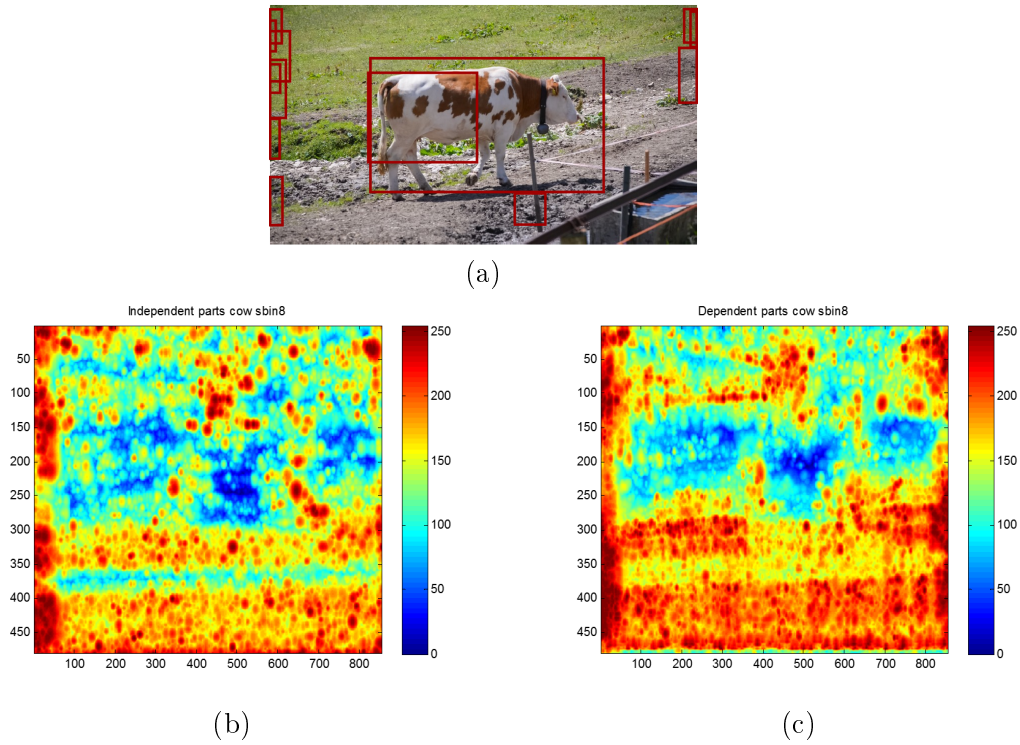


Figura 5.6: Segmentación de la secuencia vaca, (a) es un ejemplo de detección con un *threshold* en concreto, (b) representa la máscara de segmentación mediante detección basado en partes independientes y (c) la máscara de segmentación mediante detección basado en partes dependientes.

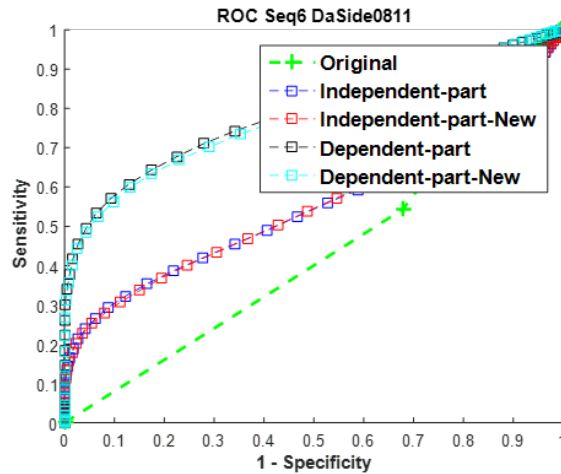


Figura 5.7: Comparativa de nuestro algoritmo (denominado en la gráfica como nuevo) con el algoritmo original para una secuencia de personas.

Capítulo 6

Conclusiones y trabajo futuro

6.1. Conclusiones

En este trabajo se planteó la tarea de realizar un segmentador objeto-fondo, partiendo del algoritmo segmentador persona-fondo descrito en [1], que fuese capaz de generar las máscaras de segmentación que segregasen los elementos deseados del resto de la imagen.

En primer lugar, se realizó un estudio del exhaustivo Estado del Arte relacionado con la tecnología necesaria en la que se basa el segmentador original. En concreto vimos los detectores que se emplean; detectores holísticos y basados en partes, y el detector DPM. Por último, describimos dicho segmentador persona-fondo.

En segundo lugar, analizamos los distintos *datasets* de segmentación y modelos de objetos de los que disponíamos para realizar una selección del conjunto de secuencias de imágenes que nos iban a ser útiles para nuestra finalidad. Vimos que al buscar imágenes que coincidiesen con los modelos, y que además los objetos de éstas imágenes guardasen las características necesarias para ser encontrados por el detector, se redujo mucho el tamaño de los *datasets* de los que partíamos.

En tercer lugar, se modificó el algoritmo original con la finalidad de ser genérico, es decir, ser capaz de segmentar una secuencia de imágenes con su respectivo modelo de objeto, y no solamente segmentar el modelo de personas. Surge así la necesidad de introducir la nueva estructura de modelos de objetos, ya que pueden tener cualquier número de poses.

Por último se evaluaron los resultados obtenidos observando las máscaras de segmentación, comparando las curvas ROC y las áreas bajo la curva resultantes de la segmentación. Los resultados muestran grupos de comportamientos diferenciados, aquellos en los que los resultados son los esperados y otros donde la segmentación

no se realizó con éxito.

6.2. Trabajo futuro

Una vez desarrollado nuestro sistema y con los resultados obtenidos en él, se propone seguir generalizando el algoritmo original para las versiones extendidas IEBP y DEBP, y la versión post-procesada DEBP-P mencionadas en el Estado del Arte de los métodos basados en partes independientes y combinación de partes que nosotros desarrollamos (IOP y DOP).

Se plantea la posibilidad de buscar más *datasets* para la posterior evaluación del sistema. En concreto de secuencias de imágenes que se puedan emplear con los modelos de objetos que disponíamos y no se usaron al no coincidir las características de los objetos o no ser éstos detectables. Se podrán también aumentar los modelos de objetos entrenando modelos específicos.

A la hora de refinar los resultados de las máscaras de segmentación, podría introducirse información adicional como puede ser el color, forma, textura, movimiento, etc.

Bibliografía

- [1] Á. García-Martín, A. Cavallaro, and J. M. Martínez, “People-background segmentation with unequal error cost,” in *Image Processing (ICIP), 2012 19th IEEE International Conference on*, pp. 157–160, IEEE, 2012.
- [2] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, “Object detection with discriminatively trained part-based models,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 9, pp. 1627–1645, 2010.
- [3] B. Leibe, A. Leonardis, and B. Schiele, “Robust object detection with interleaved categorization and segmentation,” *International journal of computer vision*, vol. 77, no. 1-3, pp. 259–289, 2008.
- [4] P. Viola and M. J. Jones, “Robust real-time face detection,” *International journal of computer vision*, vol. 57, no. 2, pp. 137–154, 2004.
- [5] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1, pp. 886–893, IEEE, 2005.
- [6] J. Pont-Tuset, F. Perazzi, S. Caelles, P. Arbeláez, A. Sorkine-Hornung, and L. Van Gool, “The 2017 davis challenge on video object segmentation,” *arXiv:1704.00675*, 2017.
- [7] F. Li, T. Kim, A. Humayun, D. Tsai, and J. M. Rehg, “Video segmentation by tracking many figure-ground segments,” in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2192–2199, 2013.
- [8] R. Girshick, P. Felzenszwalb, and D. McAllester, “Lsvm-mdpm release 4 notes,” *Apr*, vol. 21, pp. 1–4, 2010.
- [9] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes (voc) challenge,” *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, 2010.

Apéndice A

Glosario

DPM	<i>Deformable Parts Model</i>
ISM	<i>Implicit Shape Model</i>
HOG	<i>Histograms of Gradients</i>
IBP	<i>Independen Body Parts</i>
DBP	<i>Dependet Body Parts</i>
IEBP	<i>Independen Extended Body Parts</i>
DEBP	<i>Dependet Extended Body Parts</i>
DEBP-P	<i>Dependet Extended Body Parts Post-processed</i>
AUC	<i>Area Under the Curve</i>
DAVIS	<i>Densely Annotated VIdео Segmentation</i>
IOP	<i>Independent Object Part</i>
DOP	<i>Dependent Object Part</i>
ROC	<i>Receiver Operating Characteristic</i>

Apéndice B

Gráficas ROC y tabla AUC

Tabla B.1: Valores AUC para todas las secuencias y los distintos valores $fp=1,2,4, 10$ según la función Original, IOP y DOP.

	AUC Original				AUC IOP				AUC DOP			
	fp=1	fp=2	fp=4	fp=10	fp=1	fp=2	fp=4	fp=10	fp=1	fp=2	fp=4	fp=10
Car RoundAbout	0.3088	0.2280	0.1528	0.0866	0.6355	0.5245	0.4118	0.2770	0.7856	0.7097	0.6210	0.4926
Bmx Trees	0.5299	0.4189	0.3136	0.1971	0.5693	0.4468	0.3305	0.2038	0.7964	0.7020	0.5937	0.4479
Bmx Segtrackv2	0.4721	0.3720	0.2798	0.1787	0.5438	0.4252	0.3150	0.1966	0.6652	0.5549	0.4426	0.3080
Horse JumpLow	0.6088	0.5144	0.4156	0.2922	0.6892	0.5875	0.4806	0.3438	0.6868	0.5830	0.4745	0.3386
Car Turn	0.2214	0.1535	0.1015	0.0551	0.7275	0.6295	0.5207	0.3757	0.8142	0.7303	0.6274	0.4740
Bmx Bumps	0.4813	0.3758	0.2790	0.1747	0.4813	0.3758	0.2790	0.1747	0.7435	0.6487	0.5484	0.4215
Car Shadow	0.3304	0.2438	0.1710	0.0994	0.3304	0.2438	0.1710	0.0994	0.8739	0.8091	0.7253	0.5920
Drift	0.7036	0.6187	0.5283	0.4130	0.6932	0.6033	0.5110	0.3959	0.8875	0.8402	0.7830	0.6964
Motorbike	0.2214	0.1535	0.1015	0.0551	0.7275	0.6295	0.5207	0.3757	0.8142	0.7303	0.6274	0.4740
Horse JumpHigh	0.5901	0.4938	0.3955	0.2755	0.6417	0.5287	0.4133	0.2748	0.7312	0.6441	0.5524	0.4360
Motocross Jump	0.4301	0.3215	0.2285	0.1355	0.4588	0.3494	0.2530	0.1537	0.5283	0.4128	0.3069	0.1936
Cow	0.4456	0.3337	0.2339	0.1326	0.4456	0.3337	0.2339	0.1326	0.6353	0.5240	0.4745	0.3386

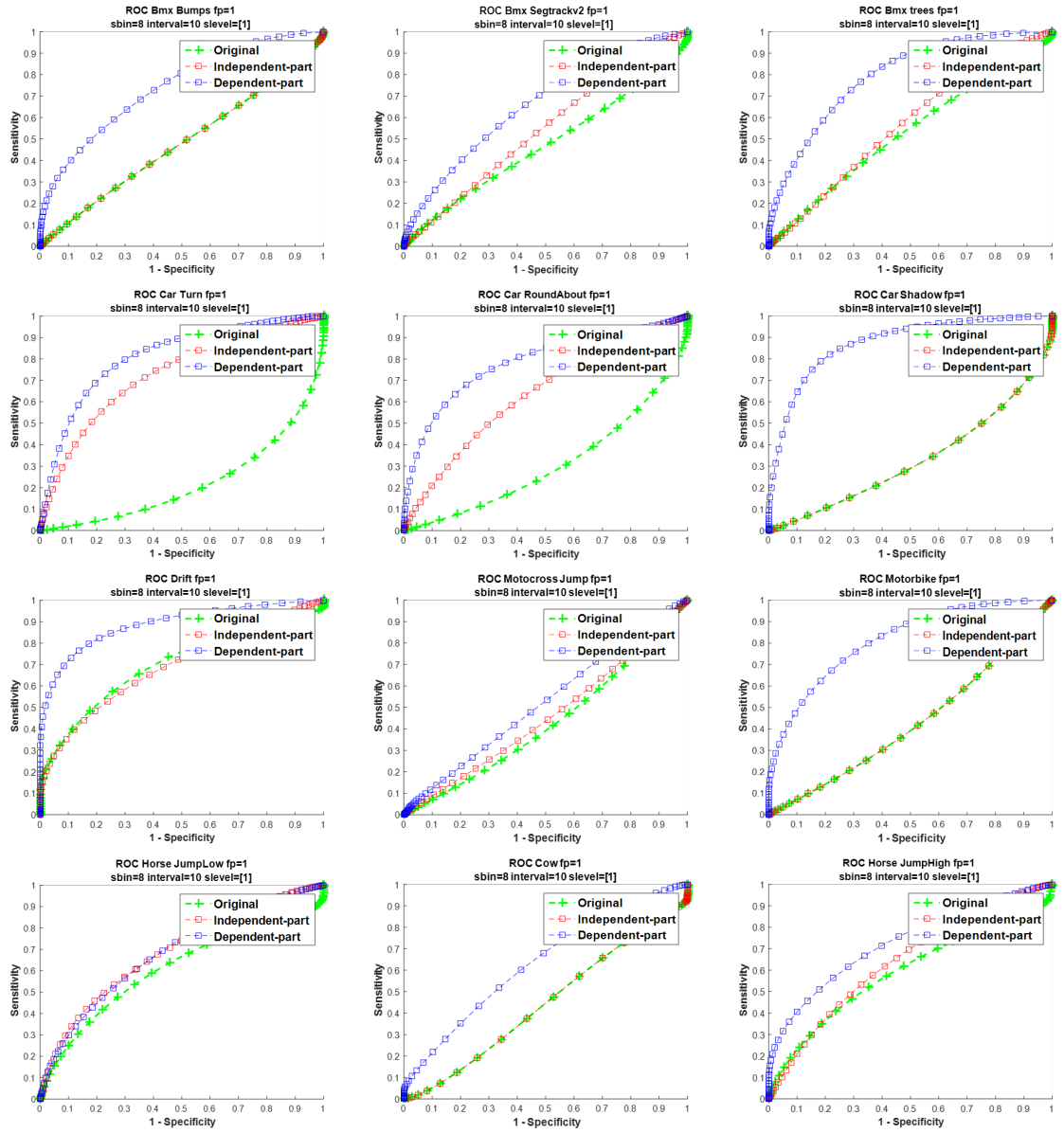


Figura B.1: Gráficas AUC de todas las secuencias para un mismo valor de falsos positivos.